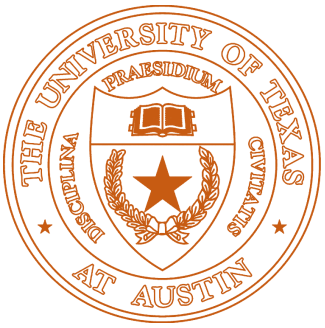
Deep Image Denoising



Shaayaan Sayed
Supervised by Professor Philipp Krähenbühl
Department of Computer Science
University of Texas at Austin

A thesis submitted for the degree of
Bachelor of Science - Turing Scholars Honors
May 2018

Abstract

Image denoising is concerned with constructing a mapping from a set of noisy images to a set of clean images. Nearly all existing denoising methods make highly restrictive assumptions about the noise which rarely apply to real-world noise distributions. In this work, we present a denoising framework that makes no assumptions about the noise structure. Inspired by existing unsupervised image-to-image translation works, we learn a denoiser network from unpaired samples of noisy and clean images. In addition to minimizing an adversarial loss, the network learns to denoise fake noisy images which are generated by sampling an estimate of the noise distribution. Our approach achieves state-of-the-art results on synthetic noise models. Additionally, we evaluate our approach on real noise by denoising raw Kinect depth images in the NYUv2 dataset and then checking for improvements in segmentation performance for indoor scene recognition. In doing so, we propose a novel denoising benchmark that can help us evaluate if denoising is useful for higher-level vision tasks.

1 Introduction

Noise is present in every real-world image capturing process. Image denoising is used to preprocess images before they are viewed by a human or used as inputs for higher level vision or graphics tasks. Denoising involves predicting a high-quality signal from a measurement that is corrupted by noise following some distribution. Noise distributions in the real world are diverse and vary depending on the setting and sensor type. For example, noise captured in a medical image taken with an X-ray sensor is structurally different from that found in a LiDAR depth image. The latter is unlike the noise present in a Kinect depth image [1] eventhough they may share the same underlying signal. Traditional approaches focus on hand-engineering pipelines to denoise a specific environment and sensor device. In nearly every case, the probabilistic form of the noise distribution is unknown. This prompts researchers to make unrealistic assumptions about the noise properties. Even when successful, these denoisers do not account for another, more intricate source of noise: inconsistency in hardware between individual sensors. In this respect, the standard framework is inflexible–any change in the image capturing process requires new assumptions and a new algorithm, developed through tedious trial and error. More recently, researchers have shifted from an assumption-based approach to a data-driven one, proposing methods that learn a denoising function from data.

Convolutional Neural Networks (CNNs) can learn complex functions for image prediction tasks such as classification, segmentation, and image translation [2, 3, 4]. In a supervised learning approach to image denoising, the denoising network can be trained to minimize a visual similarity loss between a paired dataset of noisy images and their high-quality counterparts. However, in real-world scenarios, very rarely does one deliberately capture a high and low quality measurement of the same signal. In many environments, this is not even possible. In this work, we are interested in learning denoising functions in the absence of paired training examples.

We propose an image denoising method that learns from unpaired datasets of noisy and clean images. Additionally, it makes no structural assumptions about the noise. Figure 1 illustrates our framework. We learn a denoiser for the noise distribution in a given dataset. For all cases, the training uses images of the same environment and modality but does not require the same image to exist in both clean and noisy form. The denoiser is a neural network that trains with an adversarial loss, similar to Generative Adversarial Networks (GANs) [5], learning to produce predictions that are, in principle, indistinguishable from clean images. To constrain the objective further, we introduce an additional denoising task. During training, we generate fake noisy images, for which we have the clean version for, and enforce the network to denoise them. The fake noisy images are generated by sampling from a model of the noise induced by the network and adding these noise estimates to true clean images. The process assumes that the noise is additive and independent from the underlying image content.

To evaluate our method, we test its performance on synthetically generated noise. Synthetic models, however, are not entirely representative of the noise distributions in the real-world, yet nearly all work in denoising stops here when benchmarking algorithms. As part of our evaluation, we apply our method to a dataset with real noise. Specifically, we denoise depth images of indoor scenes in the NYUv2 dataset [6], using noise-free, computer simulated scenes from SceneNet [7]. We then check for improved performance on indoor scene recognition, a task crucial for robot navigation. To the best of our knowledge, little to no recent work has been done on determining whether denoising improves the performance of higher level vision tasks.

2 Related Work

Traditional denoisers are hand engineered and exploit well-known natural image statistics. By averaging in small patches, filtering methods [8, 9] aim to preserve relatively invariable local structure while diminishing noise. Many methods perform a thresholding on the wavelet transform coefficients, removing from the image high-level frequencies which contain most of the noise [10]. The nonlocal mean algorithm [11] averages over similar patches throughout the noisy image rather than just the local neighborhood. More recently, BM3D [12] is a nonlocal, patch-based method that performs a shrinkage of the transform spectrum on groups of similar image patches. Classical methods such as these perform a mapping from noisy to clean images by exploiting an evermore complex prior over images and noise distributions. With CNNs, we can learn mappings between images automatically from data with fewer assumptions about image and noise structure.

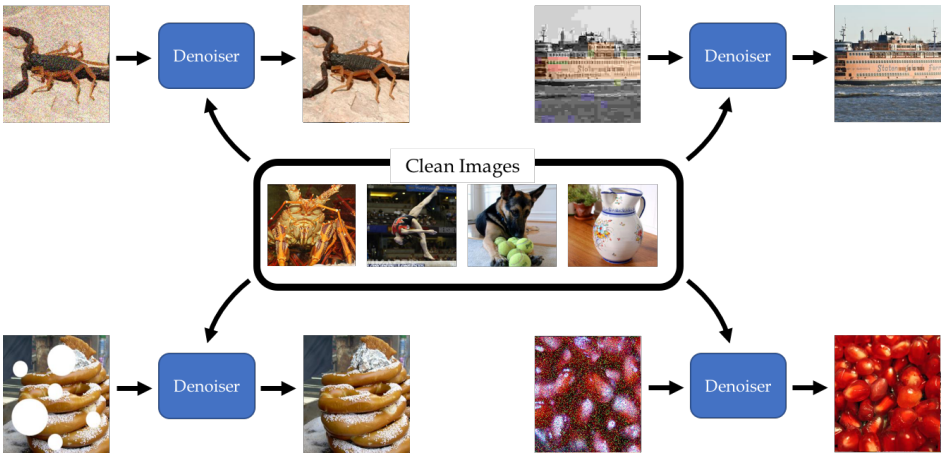


Figure 1: Our algorithm learns a scenario-specific denoiser for a given dataset corrupted by an unknown noise distribution, requiring only an additional dataset of clean images of the same environment and modality. The datasets need not be paired; they can be unordered collections. In the example given, the environment is real images and the modality is RGB. The same method can be used to learn a denoiser for different kinds of noise distributions.

The general task of learning mappings between images from data is known as image-to-image translation [4]. Image-to-image translation assumes that there is underlying structure shared by two image domains. Examples of where it can be used include converting photos to semantic labels, satellite images to maps, or noisy images to noiseless images, which is the translation that image denoising is concerned with.

There are two streams of research in supervised image-to-image translation: one that does not use adversarial training and one that does. Chen and Koltun [13] propose a multi-part CNN that gradually generates higher resolution outputs for photographic image synthesis from semantic maps. For training, they use a perceptual loss–the pretrained VGG activations of the network output must match those of the ground truth. On the other hand, many approaches train networks with an adversarial loss, which is the prominent approach to image synthesis. In the "pix2pix" framework proposed by Isola et al. [4], a GAN loss is coupled with a regression term between the network output and the ground truth. Both of these approaches require paired training examples. To learn mappings over unpaired datasets, we look to the many methods proposed for unsupervised image-to-image translation.

To learn a translation between two domains, Zhu et al. [14] suggest having mappings for both directions, training two GANs. They also introduce a cycle-consistency constraint–an image in one domain, when translated to the other and then mapped back, should be unchanged. Their setup is not fit for stochastic mappings like noise models. Still, it is influential for our work because a key component of our method is inspired by cycle-consistency.

Shrivastava et al. [15] propose an architecture to generate photo-realistic images from synthetic ones. This is desirable because instead of training networks on vast synthetic datasets, which often fail to generalize to real images, we can train them on large photo-realistic datasets generated by translating synthetic images to the real domain. Their approach pairs a GAN loss with a regression term that penalizes large changes between the input and network output in order to preserve the synthetic annotations. They apply their method to gaze-estimation and hand-pose estimation. In the context of denoising, their architecture makes the assumption that the noise is zero, which is not ideal. However, their work is an important precursor to ours because they are interested in the same question: can domain transfer aid training for other vision tasks?

Eventhough denoising methods are pervasive, few papers propose experiments for benchmarking on datasets with real noise. Recently, Plötz and Roth [16] construct a dataset of paired noisy and noise-free RGB images, taken with different ISO values in realistic settings. Even fewer papers explore the effect of denoising on higher-level vision tasks. Liu et al. [17] train a denoising network jointly with a network performing a higher-level task such as classification or segmentation. They, however, operate in the paired setting, only experimenting on synthetically generated Gaussian noise.

In the case of depth enhancement, specifically, there is a sizable body of work for depth-specific denoising. To the best of our knowledge, none conduct a quantitative evaluation after denoising real depth images. Many researchers synthetically alter clean depth images to simulate noise [18, 19]. Others use real noise images but only discuss visual results [20, 21] .

3 Preliminaries

Let $x \in X$ and $y \in Y$ be noisy and clean images, respectively. Here, X and Y refer to the possibly infinite set of noisy and clean images. They follow the data distributions given by $p_X(x)$ and $p_Y(y)$. We assume there exists an additive noise model which relates X and Y. In general, $|X| \gg |Y|$, multiple noisy images $\{x_0^k, x_1^k, \ldots\}$ might correspond to a single clean image y_k. The noise model is formulated as:

$$x_i^k = y_k + n_i \tag{1}$$

where n_i is noise (defined as an image) sampled from some unknown noise distribution $p_N(n)$. Note that $p_X(x)$ can be written as a convolution of p_Y and p_N: $p_X(x) = \iint_{x=y+n} p_Y(y)p_N(n)\,dy\,dn$. Similarly, we can rewrite the set of noisy images as $X = \{y + n \mid y \in Y, n \in N\}$. An important assumption we make is that clean images and image noise are independent, $(y, n) \sim p_Y(y)p_N(n)$. Extensions to a multiplicative noise model is trivial, as each datum could undergo a log-transform. Dependent models are beyond the scope of this work.

We are interested in constructing a mapping $F : X \to Y$ using unpaired samples from p_x and p_y. Such a mapping F is a denoiser that removes noise from images in X. This can be formulated as a conditional generative modeling task where we wish to estimate the true conditional $p_{Y|X}(y|x)$. We parametrize F as a neural network. Our goal is to learn the optimal denoiser F^* which satisfies :

$$F^*(y + n) = y \tag{2}$$

for $y \sim p_y(y), n \sim p_n(n)$.

As a start, we can enforce the output of F to resemble the set of clean images. Let p_F be the distribution of denoised images $F(X)$. We would like $p_F \approx p_Y$. We can learn p_Y by using the Generative Adversarial Network (GAN) framework [5].

GANs minimize the Jensen-Shanon Divergence $JSD(P \parallel Q)$ between two probability distributions P and Q. $JSD(P \parallel Q)$ is zero if $P = Q$ and strictly positive otherwise. GANs comprise of a generator network producing samples of Q and a discriminator network D distinguishing samples of the generator from P. They are optimized as a minimax two player game–the generator tries to fool the discriminator into believing it produces true samples from P, while the discriminator constantly updates its model. If we treat the denoising model F as the generator network, we can use an adversarial loss to minimize $JSD(p_F \parallel p_Y)$, pulling the distribution of denoised images close to the distribution of clean ones. Formally, the objective is given by (see [5] for JSD formulation)

$$L_{GAN}(F) = \max_D \; \mathbb{E}_{y \sim P_Y}[logD(y)] + \mathbb{E}_{x \sim P_X}[log(1 - D(F(x)))] \tag{3}$$

In practice, we have F maximize $\mathbb{E}_{x \sim P_X}[log(D(F(x)))]$ to prevent saturated gradients early in training.

Unfortunately, the adversarial loss by itself is insufficient. There are infinitely many mappings that can approximate p_Y, including any permutation of the true denoising function [22]. To ensure that a meaningful relationship between noisy images and clean predictions (one that resembles Eq. 1) is learned, we need to further constrain F. There are many popular constraints in current work.

SimGAN [15] assumes that F is close to identity. This implies $n \approx 0$ for each noise sample drawn. Note, this is stricter than the common zero mean noise assumption and encourages every noise sample to be zero.

Another popular constraint is assuming F is invertible [14]. The CycleGAN approach implements this by learning the noise model through a second mapping $G : Y \to X$. Similar to F, G minimizes $JSD(p_G \parallel p_X)$ where p_G is the distribution of corrupted images over $G(Y)$. To enforce that F and G are inverses, both networks additionally minimize two cycle-consistency losses which ensure $G(F(x)) \approx x$ and $F(G(y)) \approx y$. CycleGAN has two flaws when used for image denoising. First, G is implemented as a deterministic mapping, which is not true for any noise model. Second,

$G(F(x)) \approx x$ forces G to be a one-to-one mapping when cycling $X \to Y \to X'$. This implies there is just a single noise pattern for each clean image, or worse there is just a single global noise pattern for independent noise.

One straightforward way to reform CycleGAN is to make G stochastic and drop the $G(F(x)) \approx x$ constraint. G can be made stochastic by defining it as $G : Y \times Z \to X$ where Z is a latent space with a Gaussian prior $p_z(z)$ over its elements. The mapping takes as input a vector of noise and a clean image sample and generates a noisy image. Note, under a stochastic G, the constraint in question now becomes $G(F(x), z) \approx x$ for $z \sim p_z$. This is still incorrect as G learns to map $F(x)$ to the same x regardless of z. Even without this constraint, G ignores z in practice, failing to produce diverse outputs [4]. Under this modified CycleGAN setup, $G(y)$ is a poor estimate for the conditional distribution of noisy images given a clean image $p_{X|Y}(x|y)$. In the next section, we consider an alternative method to estimate $p_{X|Y}$.

4 Approach

In order to learn a mapping from noisy to clean images given unpaired samples from p_X and p_Y, we train F to minimize a combination of two losses:

$$L(F) = L_{GAN}(F) + \lambda L_{reg}(F) \tag{4}$$

where L_{reg} is a regularizing constraint that helps F learn a useful relationship between X and Y. Ideally, we would like $L_{reg} = \mathbb{E}_{y \sim p_y(y), x \sim p_{X|Y}(x|y)}[\|F(x) - y\|_1]$. This represents the supervised training scenario similar to pix2pix [4], where we have a clean image and multiple corresponding noisy images, and F minimizes an L1 loss between pairs. Unfortunately, we don't have access to $p_{X|Y}(x|y)$. We propose to estimate $p_{X|Y}(x|y)$ by first drawing samples from an estimate of the noise distribution $p_{\tilde{N}}$ and adding those samples to true clean images $y \sim p_Y$. This directly follows the noise model in eqn. 1, except we sample the noise from $p_{\tilde{N}}$ because we don't know p_N.

Given a pair of unrelated noisy and clean images (x_j, y_k) we generate a noise estimate and a fake noisy image ($\tilde{n_j}$, $\tilde{x_j^k}$) as follows:

$$\tilde{n_j} = x_j - F(x_j), \quad \tilde{x_j^k} = y_k + \tilde{n_j} \tag{5}$$

We argue that F should be able to denoise $\tilde{x_j^k}$ back to y_k: $F(y_k + \tilde{n_j}) = y_k$, directly following the optimal denoiser F^* in eqn. 2. We incentivize this by letting the regularization loss in eqn. 4 be:

$$L_{reg}(F) = \mathbb{E}_{x \sim P_X, y \sim P_Y}[\|F(y + x - F(x)) - y\|_1] \tag{6}$$

Our training objective makes no assumptions about the structure of the noise, but it assumes independence between the image and noise. In practice the model is even able to handle a moderate amount of image dependent noise, as we will show in Section 5. We also don't make any assumptions about the denoising function F other than it being a deep network.

One way to interpret our approach is F minimizes a cycle-consistency loss during $Y \to X \to Y'$. Instead of learning a mapping for the noise model $Y \to X$ like CycleGAN, we formulate an estimator for $p_{X|Y}(x|y)$.

It is trivial to see that the optimal denoiser F^* (from eqn. 2) is a global minimum for L_{GAN} (eqn. 3) and L_{reg} (eqn. 6). Goodfellow et al. [5] showed that L_{GAN} is at a global minimum if and only if $p_F = p_Y$. It is obvious that p_{F^*} (the distribution of denoised outputs from the optimal denoiser $F^*(X)$) is equal to p_Y. For any $y \sim p_Y, n \sim p_N$, $F^*(y + n) = y$, which is distributed according to p_Y. The following proof shows F^* is a minimum for L_{reg}:

$$\begin{aligned}
L_{reg}(F^*) &= \mathbb{E}_{x \sim p_X, y \sim p_Y}[\|F^*(y + x - F^*(x)) - y\|_1] \\
&= \mathbb{E}_{y_1 \sim p_Y, n \sim p_N, y_2 \sim p_Y}[\|F^*(y_2 + (y_1 + n) - F^*(y_1 + n)) - y_2\|_1] \\
&= \mathbb{E}_{y_1 \sim p_Y, n \sim p_N, y_2 \sim p_Y}[\|F^*(y_2 + (y_1 + n) - y_1) - y_2\|_1] \\
&= \mathbb{E}_{y_1 \sim p_Y, n \sim p_N, y_2 \sim p_Y}[\|F^*(y_2 + n) - y_2\|_1] \\
&= \mathbb{E}_{y_1 \sim p_Y, n \sim p_N, y_2 \sim p_Y}[\|y_2 - y_2\|_1] \\
&= 0
\end{aligned}$$

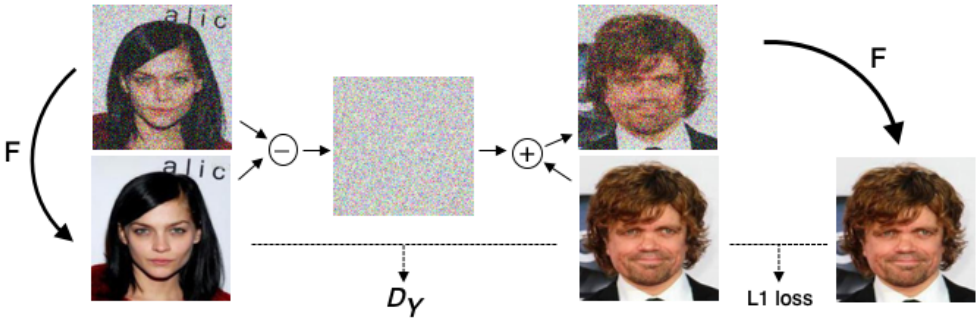


Figure 2: Our approach learns a denoising mapping F by minimizing a combination of an adversarial loss and a regularization term. F takes noisy images as input and outputs denoised images. It is trained to fool a discriminator network D_Y which learns to distinguish between denoised images from F and real clean images. F must also learn to denoise corrupted clean images, enforced by a regularization term. These fake noisy images are generated by sampling noise estimates and adding them to true clean images.

We were unable to come up with a concise set of constraints that make the optimum unique, however experimentally the method converges close to the optimal denoiser, obtained from a paired dataset of noisy and clean images.

Network Architecture For the generator, we use a residual network with 6 ResNet Blocks. We also experiment with U-Net, but found ResNet to be better for image denoising. Both architectures are widely used in image generation tasks. The discriminator is a 60 x 60 PatchGAN. It classifies at the scale of patches, allowing the network to better capture high-level frequencies.

Training Details We use two standard approaches to stabilize GAN training. First, in L_{GAN} we replace the negative log likelihood objective with a least squares loss. Specifically, we train F to minimize $\mathbb{E}_{x\sim P_X}[(D(F(x)-1)^2]$, and we train D to minimize $\mathbb{E}_{y\sim P_Y}[(D(y)-1)^2]+\mathbb{E}_{x\sim P_X}[D(F(x))^2]$. Least Squares GANs (LSGANs) [23] have been shown to produce higher quality results. Secondly, we feed the discriminator samples from the generator's output history rather than just the latest fake outputs. We sample from a buffer that stores the 50 latest generator outputs. We set $\lambda = 10$ in eqn. 4 for all trials. Furthermore, all networks are trained with an Adam solver with learning rate of 0.0002. We closely follow the implementation and training procedures found in [14, 24].

5 Results

We first evaluate our approach on synthetic noise models where noisy-clean image pairs are available. Second, we apply it to denoise depth images for indoor scene understanding. We compare our approach to other traditional denoising methods and recent image-to-image translation frameworks.

5.1 Evaluation

Testing denoising methods on images with synthetic noise allows for straightforward evaluation because we have access to the true clean image. We use two datasets: Cityscapes [25] and ImageNet [26]. To simulate the unpaired denoising setting, we divide the dataset into two, equal-sized, non-overlapping subsets where one is corrupted with noise and the other is untouched. For the metrics, we use peak signal to noise (PSNR) and structural similarity (SSIM) [27].

denoiser	Gaussian		Salt and Pepper		JPEG Compression	
	psnr	ssim	psnr	ssim	psnr	ssim
CBM3D [28]	29.906	0.851	10.959	0.064	26.892	0.793
SemiCycleGAN	11.114	0.227	14.187	0.374	25.676	0.767
CycleGAN [14]	10.99	0.212	12.886	0.279	10.901	0.315
SimGAN [15]	29.748	0.811	**33.042**	0.888	27.18	0.756
ours	**31.032**	**0.867**	32.831	**0.918**	**27.685**	**0.796**
pix2pix [4]	31.599	0.873	39.944	0.98	28.119	0.814

Table 1: Denoising results for the CityScapes test set, given by peak signal-to-noise ratio (psnr) and structural similarity index (ssim). Three noise models are applied: additive gaussian, salt and pepper, and jpeg compression. Pix2pix results are shown as an upper bound reference using fully supervised training.

denoiser	Gaussian		Salt and Pepper		JPEG Compression	
	psnr	ssim	psnr	ssim	psnr	ssim
CBM3D [28]	**26.645**	**0.767**	20.104	0.411	**24.832**	**0.708**
SemiCycleGAN	16.741	0.549	6.381	0.001	14.909	0.474
CycleGAN [14]	19.412	0.62	6.588	-0.012	21.881	0.624
SimGAN [15]	24.808	0.65	**25.412**	**0.702**	23.448	0.652
ours	22.469	0.627	22.534	0.662	23.384	0.662
pix2pix [4]	27.224	0.78	26.898	0.831	23.512	0.634

Table 2: Denoising results for the ImageNet test set.

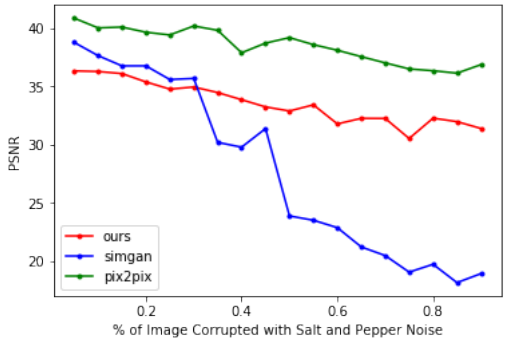


Figure 3: PSNR results on CityScapes validation set as amount of salt and pepper noise increases.

Denoiser	Not pretrained			SceneNet pretrained		
	pix. acc.	class acc.	class IOU	pix. acc.	class acc.	class IOU
None	.568	.408	.268	.654	.537	.374
colorization [29]	.563	.408	.270	**.673**	**.556**	**.397**
simGAN [15]	.487	.330	.205	.569	.442	.296
ours	.496	.336	.211	.572	.443	.296

Table 3: Semantic segmentation scores on NYUv2 using different denoisers for depth image processing. Segmentation models are trained with random initialization and also pretrained weights using SceneNet. Metrics used for evaluation are pixel accuracy, mean class accuracy, and mean class intersection over union [25].

We chose diverse noise models with varying structural properties to test the general applicability of our method. For each dataset we evaluate three noise models: additive Gaussian noise, salt and pepper noise, and JPEG compression artifacts. We chose JPEG compression to include a dependent noise model. Note that out of the three, only one model is additive.

Cityscapes Cityscapes contains 2975 training, 500 validation, and 1525 test images. We corrupt 1487 images with noise and leave the other 1488 as the clean dataset. We train for 200 epochs, decaying the learning rate to zero starting from the 100th epoch.

ImageNet Evaluating on ImageNet exhibits performance on larger, more diverse scenarios. We train and test on a subset of ImageNet–we sampled five images for each of the 200 categories in the ILSVRC2017 object detection challenge, resulting in 9796 train (4897 noisy and 4899 clean), 200 val, and 2502 test images. The models were trained for 6 epochs, and the learning rate was decayed to zero starting from the 3rd epoch.

5.1.1 Baselines

CBM3D [28] is an extension of BM3D [12] for color images. It is the state of the art non-deep denoiser. We use a publicly released implementation[1].

CycleGAN [14] learns an inverse mapping $G : Y \to X$, in addition to $F : X \to Y$. It applies adversarial losses to both. Two cycle consistency losses ensure the two functions invert each other $F(G(Y)) \approx Y$ and $G(F(X)) \approx X$. For denoising, G aims to replicate the noise model. Unfortunately, the true noise model not deterministic and is poorly approximated by a fixed network G. In our experiments, we found that enforcing $G(F(X)) \approx X$ almost always results in failure. Therefore, we only apply the forward cycle loss, $\|F(G(Y)) - Y\|_1$. We call this baseline **SemiCycleGAN**.

SimGAN [15] learns a mapping $F : X \to Y$ using an adversarial loss. In addition, the denoiser minimizes a pixel-wise loss between the noisy image and denoised output denoted as $\|F(X) - X\|_1$.

pix2pix [4] is a paired image-to-image translation algorithm. We use it as an oracle method giving us the performance of the best network if the noise model is known.

5.1.2 Comparison against baselines

On the Cityscapes task, as can be seen in Table 1, our method outperforms other baselines in nearly every trial. For the Gaussian and JPEG experiments, it produces outputs that are comparable in quality to the supervised pix2pix. The outputs of our approach for these distributions are sharper and more detailed than those of simGANs, shown in fig. 5.

On ImageNet, however, our method does not perform as well. The decrease in performance from Cityscapes to ImageNet is seen with simGAN and pix2pix as well. We suspect this is because trained GANs are known to learn distributions of low support, a problem known as mode collapse [30]. This is especially problematic in a highly diverse dataset like ImageNet. See visual results in fig. 6.

For both datasets, semiCycleGAN and CycleGAN fail to produce compelling results. Furthermore, CBM3D does not perform well on the Salt and Pepper and JPEG compression experiments. This is expected because it was designed for additive white gaussian noise [31].

SimGAN performs well on salt and pepper noise for both datasets. This may be because a large fraction of the pixels are left uncorrupted with this noise model, which is ideal for the simGAN assumption that the noisy image and denoised output should be similar. In fig. 3, we evaluate simGAN and our approach on increasing salt and pepper noise levels. SimGAN's performance drops considerably when higher fractions of the image is corrupted. Our method also decreases in effectiveness, but its drop in performance is comparable to pix2pix's progression.

5.2 Application

Adequate performance on synthetic models shows proof of concept, but it does little to indicate how the method may perform on real-world noise distributions. Therefore, we apply our method to denoise raw depth images in the NYUv2 [6] dataset. NYUv2 is used for learning indoor scene recognition, and it contains 795 train and 654 test images of diverse indoor scenes captured with an

[1] `www.cs.tut.fi/~foi/GCF-BM3D/`

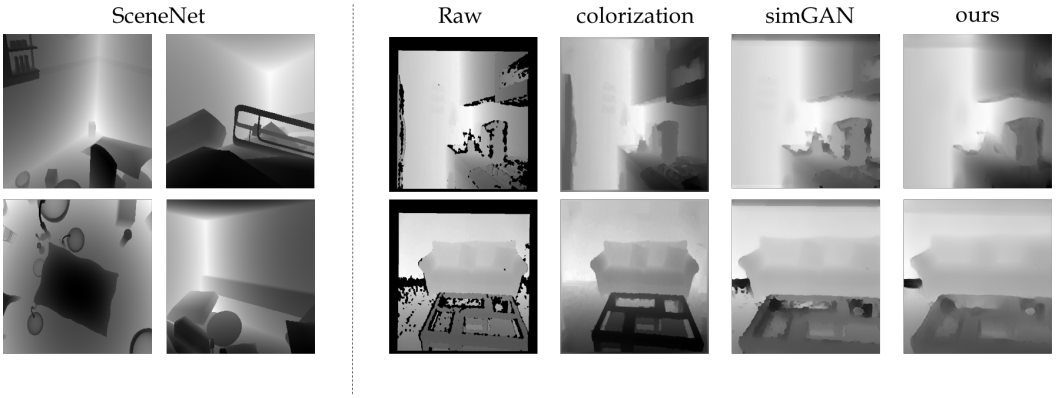


Figure 4: Example denoised depth images on NYUv2 indoor scene recognition dataset. (Left) noise-free images from SceneNet, (right) comparison across denoisers on noisy depth images

RGB-D structured-light Kinect camera. The depth, specifically, is measured with a near-infrared sensor, which is susceptible to noise caused by the scattering of near-infrared rays on low-reflectance surfaces and object boundaries [32]. This noise results in missing depth values, which can be seen as blacked-out object boundaries in fig. 4. Sometimes, entire objects like mirrors can be missing. We obtain noise-free depth scenes from SceneNet RGB-D [7], a dataset of 5M computer simulated indoor scenes.

Because paired clean and noisy depth images don't exist, it is unclear how to measure performance. In this work, we test whether denoising raw depth images in NYUv2 improves semantic segmentation accuracy of indoor scenes. By this criteria, a well-performing denoising algorithm is one that predicts useful signals for subsequent feature learning.

We compare our method to simGAN and also to the colorization scheme proposed in [29], which fills in missing depth values using the depth measurements of neighboring pixels with similar intensities in YUV space. The latter method is the standard preprocessing technique for raw depth images in NYUv2.

Training Details The denoising network is trained using the same architecture and hyperparameters as section 4. We train it for 100 epochs, starting the learning rate decay at epoch 50. For the segmentation network, we use a U-Net architecture. We experiment with pretraining the network on SceneNet, which has been shown to outperform randomly initialized weights and networks pretrained on ImageNet [7]. We train the network on NYUv2 for 100 epochs with a learning rate of 0.05, decaying the rate to zero from the 50th epoch.

5.2.1 Discussion

As shown in Table 3, both simGAN and our approach fail to improve segmentation performance on NYUv2. In fact, they hinder it. Visual results are shown in fig 4. While simGAN and our approach diminish noise artifacts in raw images, they may just be smoothing over the depth space. This is not ideal for depth enhancement. Noise-free areas may be averaged with bordering noise, resulting in blurred details and loss of clean signal.

On another note, denoising by colorization only improves performance when the network is pretrained on SceneNet. Without pretraining, it has no effect as a preprocessing technique, performing as well as the noisy, raw depth images.

6 Limitations and Conclusion

We present a denoising method that is inspired from existing unsupervised image-to-image translation architectures. Our method achieves state-of-the-art results on synthetic noise models for the CityScapes dataset, beating the well-known BM3D algorithm. The results are not as good for the ImageNet dataset, but this may be a fault of the Generative Adversarial Network framework we used.

Much work is being done in developing more stable GANs that can better learn diverse and large data distributions. This research will only improve translation methods like ours.

We also conduct a quantitative experiment on real noise, testing whether denoising depth images improves the segmentation accuracy of indoor scenes. To the best of our knowledge, this is a novel benchmark for denoising algorithms. Unfortunately, our method does not improve performance, but it does produce visually cleaner outputs. One interesting finding is that the colorization algorithm, which comes with NYUv2 to denoise depth images, improves performance only when the network is pretrained on SceneNet. This is encouraging because it shows denoising can improve performance of higher-level vision tasks, but it is unclear what modalities or network conditions allow for this. We hope this work starts to shift the potential of denoising algorithms from simply being a tool to produce visually pleasing images to an important preprocessing technique for high-performing vision pipelines.

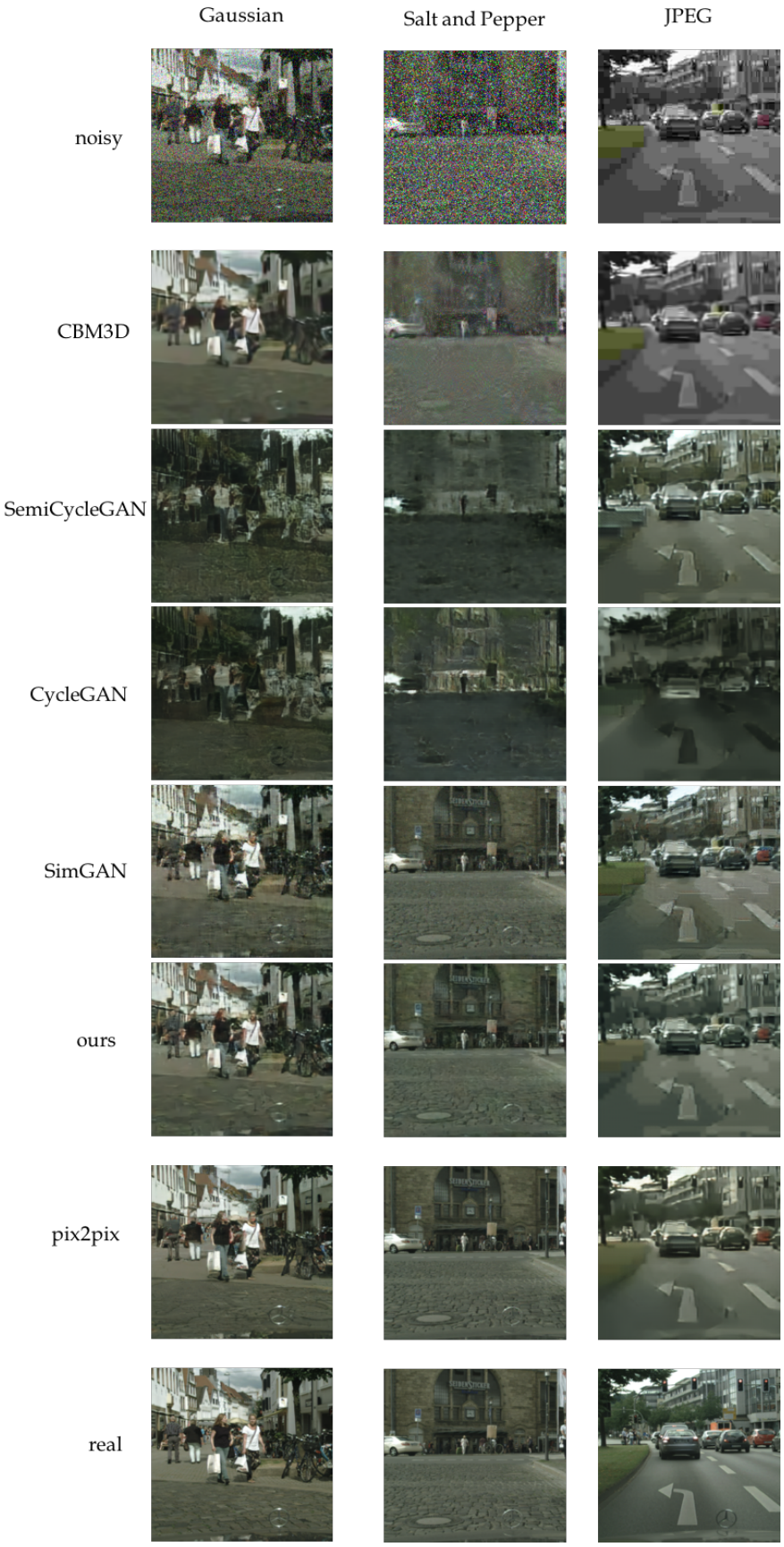


Figure 5: Visual comparison of baselines on CityScapes

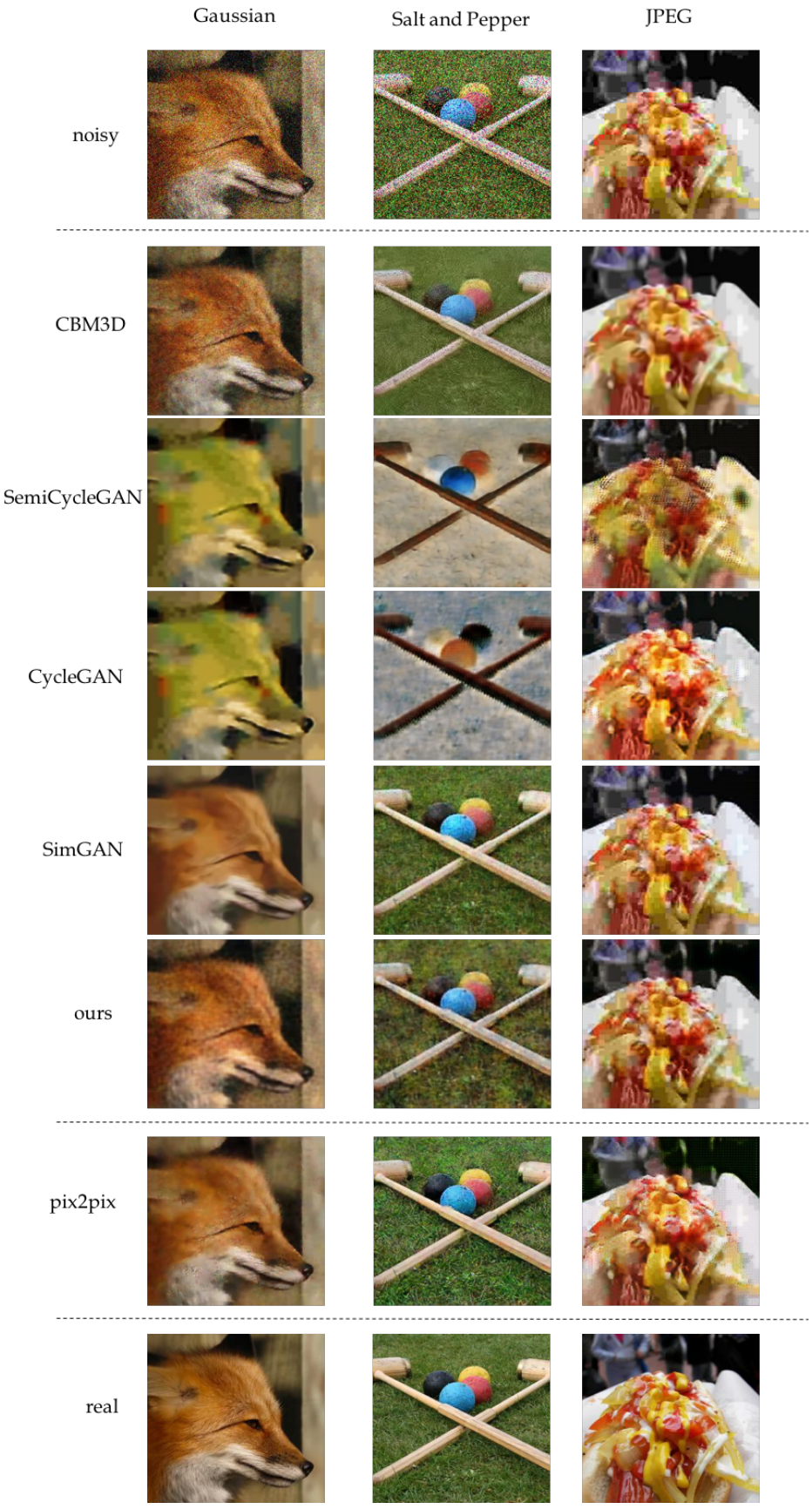


Figure 6: Visual comparison of baselines on ImageNet

References

[1] Martin Hämmerle, Bernhard Höfle, and Johannes Fuchs. Comparison of kinect and terrestrial lidar capturing natural karst cave 3-d objects. *IEEE Geoscience and Remote Sensing Letters*, 11:1896–1900, 2014.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL `http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`.

[3] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1605.06211, 2016. URL `http://arxiv.org/abs/1605.06211`.

[4] P. Isola, J. Zhu, T. Zhou, and A. Efros. Image-to-image translation with conditional adversarial nets. *CVPR*, 2017.

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NIPS*, 2014.

[6] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[7] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J.Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? 2017.

[8] A. Lev, S. Zucker, and A. Rosenfeld. Iterative enhancement of noisy images. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(6):435–442, 1977.

[9] M. Lindenbaum, M.Fischer, and A. Bruckstein. On gabor's contribution to image enhancement. *Pattern Recognition*, 27(1):1–8, 1994.

[10] D. Donoho. Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data. *In Proceedings of Symposia in Applied Mathematics*, pages 173–205, 1993.

[11] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. *CVPR*, 2: 60–65, 2005.

[12] K. Dabov, A. Foi, V. Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions On Image Processing*, 16(8):2080–2095, 2007.

[13] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. *CoRR*, abs/1707.09405, 2017. URL `http://arxiv.org/abs/1707.09405`.

[14] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, 2017.

[15] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. *arXiv:1612.07828v2*, 2014.

[16] T. Plötz and S. Roth. Benchmarking denoising algorithms with real photographs. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2750–2759, July 2017. doi: 10.1109/CVPR.2017.294.

[17] Ding Liu, Bihan Wen, Xianming Liu, and Thomas S. Huang. When image denoising meets high-level vision tasks: A deep learning approach. *CoRR*, abs/1706.04284, 2017. URL `http://arxiv.org/abs/1706.04284`.

[18] S. Lu, X. Ren, and F. Liu. Depth enhancement via low-rank matrix completion. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3390–3397, June 2014. doi: 10.1109/CVPR.2014.433.

[19] Jennifer Dolson, Jongmin Baek, Christian Plagemann, and Sebastian Thrun. Upsampling range data in dynamic environments. In *CVPR*, pages 1141–1148. IEEE Computer Society, 2010. ISBN 978-1-4244-6984-0. URL http://dblp.uni-trier.de/db/conf/cvpr/cvpr2010.html#DolsonBPT10.

[20] Huayang Li, Dehui Kong, Shaofan Wang, and Baocai Yin. Hand depth image denoising and superresolution via noise-aware dictionaries. In *Journal of Electrical and Computer Engineering 2016*, 2016.

[21] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. Lidarboost: Depth superresolution for tof 3d shape scanning. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 343–350, June 2009. doi: 10.1109/CVPR.2009.5206804.

[22] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. *CoRR*, abs/1703.00848, 2017. URL http://arxiv.org/abs/1703.00848.

[23] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, and Zhen Wang. Multi-class generative adversarial networks with the L2 loss function. *CoRR*, abs/1611.04076, 2016. URL http://arxiv.org/abs/1611.04076.

[24] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 465–476. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6650-toward-multimodal-image-to-image-translation.pdf.

[25] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016. URL http://arxiv.org/abs/1604.01685.

[26] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[27] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IIEEE Transactions on Image Processing*, 13:600–612, 2004.

[28] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Color denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. *ICIP*, 2007.

[29] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM Trans. Graph.*, 23(3):689–694, August 2004. ISSN 0730-0301. doi: 10.1145/1015706.1015780. URL http://doi.acm.org/10.1145/1015706.1015780.

[30] Sanjeev Arora and Yi Zhang. Do gans actually learn the distribution? an empirical study. *CoRR*, abs/1706.08224, 2017. URL http://arxiv.org/abs/1706.08224.

[31] Zhang L. Zhang D. Xu J., Ren D. Patch group based bayesian learning for blind image denoising. In *Computer Vision – ACCV 2016 Workshops. ACCV 2016.*, 2017.

[32] S. Kim, M. Kim, and Y. Ho. Depth image filter for mixed and noisy pixel removal in rgb-d camera systems. *IEEE Transactions on Consumer Electronics*, 59(3):681–689, 2013.